DOCUMENT RESUME

ED 275 307 IR 012 367

AUTHOR Oberlin, Lynn
TITLE Writing Questions and Answers with a Word Processor for Student Computer Use.
PUB DATE 23 Jan 86
NOTE 9p.
PUB TYPE Guides - Classroom Use - Guides (For Teachers) (052) -- Computer Programs (101)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Material Development; Media Adaptation; *Microcomputers; *Teacher Developed Materials
IDENTIFIERS *Apple II; *BASIC Programing Language

ABSTRACT

This paper shows how questions and answers written on a word processor can be used with a short BASIC program to present computer material which is self-correcting and self-scoring. This procedure provides a way for teachers who have little or no knowledge of computers to prepare instructional programs for students. The process for creating the questions and answers on a word processor consists of nine simple steps, which are included. The questions and answers are then saved as a text file (a few word processors save the material as binary files but provide a way to change them into text files). With the short BASIC program, the teacher now has computer material which the students can use. A sample program is provided for the Apple II family of microcomputers, together with instructions for adding routines, and the use of multiple-choice questions is discussed. (DJR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

WRITING QUESTIONS AND ANSWERS WITH A WORD PROCESSOR FOR STUDENT COMPUTER USE

LYNN OBERLIN
COLLEGE OF EDUCATION
UNIVERSITY OF FLORIDA

F.I.C.C.
ORLANDO, FLORIDA
JANUARY 23, 1986

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Lynn Oberlin

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED275307

IR012367

Writing Questions and Answers with a Word Processor

For Student Computer Use

Lynn Oberlin
College of Education
University of Florida

This presentation will show how questions and answers written with a word processor can be used with a short BASIC program to present material, on the computer, which is self correcting and self scoring. Simple instructions will be given with handouts which include BASIC programs. Ways of adding routines and the use of multiple-choice questions will be covered.

These instructions provide a way for teachers who have little or no knowledge of computers or programming to prepare instructional material for use by individual students. The following procedure is used in creating the question and anwswer file:

1. Boot up your word processor
2. Be sure the word processor is ready to start a new file
3. Write a question (questions which can be answered with one word are best)
4. Press the return key
5. Write the answer
6. Press the return key
7. Repeat steps 3 - 6 for each item
8. Type "END"
9. Save your file

The material must be saved as a text file by the word processor. (A few word processors save the material as a binary file but then usually provide a program to change it to a text file. Bank Street Writer is an example.) I usually use Applewriter II to create my text file.

The text file of questions and answers can be read into memory by an Applesoft BASIC program. An example of a simple no frills program is listed below.

```
900  REM **********************
901  REM *                      *
902  REM *    QUESTIONS &       *
903  REM *      ANSWERS         *
904  REM *                      *
905  REM *   BASIC PROGRAM      *
915  REM *  BY LYNN OBERLIN     *
917  REM *     (C) 1986         *
918  REM *                      *
920  REM **********************
999 :
1000  REM ======================
1002  REM VARIABLES
1004  REM  NU = NUMBER OF QUES
1006  REM   N = PRINTED QUES NO
1008  REM  Q$ = QUESTIONS
1010  REM  A$ = ANSWERS
1012  REM SA$ = STUDENTS ANSWER
1050  REM ======================
1099 :
1200  TEXT : HOME
1300  REM * INITALIZE *
1310  DIM Q$(30),A$(30)
1320  D$ =  CHR$ (4)
1321  REM * GET FILE *
1322  INPUT "FILE NAME =======> ";F$
1325  PRINT D$"VERIFY"F$
1330  PRINT D$"OPEN"F$
1340  PRINT D$"READ"F$
1400  I = I + 1: INPUT Q$(I): IF Q$(I) <  > "END" THEN  INPUT
      A$(I): GOTO 1400
1405  NU = I - 1
1410  PRINT D$"CLOSE"
1440  REM * START OF PROGRAM *
1450  HOME
1500  FOR N = 1 TO NU
1900  PRINT N;". ";Q$(N)
2800  PRINT : INPUT "ANSWER ======> ";SA$
3100  IF SA$ = A$(N) THEN  PRINT : PRINT  TAB( 17);"CORRECT": GOTO
      3250
3200  PRINT : PRINT  TAB( 18);"WRONG"
3250  VTAB 24: HTAB 11: PRINT "ANY KEY TO CONTINUE";
3275  GET K$
3300  HOME : PRINT : NEXT N
3450  HOME
3500  VTAB 15: HTAB 17: PRINT "GOOD-BYE"
3550  VTAB 18: HTAB 13: PRINT "HAVE A GOOD DAY"
```

In line 1310 two arrays are initalized for up to thirty items each. Q$ is for the questions and A$ is for the answers. If more questions are desired, a larger number could be used. Line 1320 sets D$ equal to control-D which is expressed as CHR$(4).

BEST COPY AVAILABLE

Next is the task of getting the file into the computer's memory. The steps used in lines 1322 - 1340 are standard procedures for reading text files. Each time through line 1400, a check is made for the word "END". If not found, both the Q$ and the A$ arrays are read into memory and line 1400 is repeated. When "END" is found, line 1405 corrects the count and line 1410 closes the file.

The program really starts at line 1450 and prints a number, the question, and asks the student for the answer. The student is told either "CORRECT" or "WRONG" and the program continues until all the questions have been used. The student is then given a "GOOD-BYE" message as the program ends.

While this program runs well, the following items are listed with suggestions for improvements.

1. Applesoft's INPUT command prevents the use of any commas or semicolons in your material and in the student's answer. This can be overcome by adding a INPUT ANYTHING routine. Below is a machine language program which can be added and easily interfaces with the BASIC program.

```
9998  REM INPUT ANYTHING
9999  REM CALL INPT,ST$
10000 INPT = 768: FOR J = INPT TO INPT + 27: READ K: POKE J,K:
      NEXT : DATA
      32,190,222,32,227,223,32,108,221,133,133,132,134,32,111,25
      3,32,57,213,200,162,141,32,233,227,76,154,218
10001 RETURN
```

This will require the changing of all INPUT statements, such as

```
10 Input A$
```

to

```
10 CALL INPT,A$
```

2. Words starting near the right side of the screen may be split. A whole word subroutine can be added to correct this problem. To use this routine, a string is changed to X$ and the command GOSUB 9000 is issued.

```
8990  REM WHOLE WORD PRINTER
9000  D =  PEEK (36): IF  LEN (X$) < 40 - D THEN 9050
9010  C = 39 - D
9020  IF  ASC ( MID$ (X$,C,1)) = 32 OR  ASC ( MID$ (X$,C,1)) =
      45 THEN 9040
9030  C = C - 1: GOTO 9020
9040  PRINT  LEFT$ (X$,C):X$ =  RIGHT$ (X$, LEN (X$) - C): GOTO
      9000
9050  PRINT X$: RETURN
```

3. Questions always appearing in the same order tend to cause students to lose interest. This is overcome by a routine which will select the questions in a random order. The following lines will do the job quite well but may sometimes cause a slight delay as larger numbers of questions are used. This should not cause a problem. However, if it is annoying, it can be overcome with a more sophisticated routine.

```
1550  REM SELECT IN RAMDOM ORDER
1600  RQ =  INT ( RND (1) * NU) + 1
1700  IF QU(RQ) = 1 THEN 1600
1800  QU(RQ) = 1
```

4. The original short program requires the student to complete all questions before stopping or to bomb out of the program in some way. I believe that a program should be under the control of the user. The two lines below give the student a choice after each question.

```
3250  VTAB 24: HTAB 2: PRINT "ANY KEY TO CONTINUE --  'Q' TO
      QUIT. ";
3275  GET K$: IF K$ = "Q" THEN 3450
```

5. A good program should "Keep score" and let the student know the results even if the student elects not to finish every item. Adding the following lines will do this. Notice that a check is made to determine whether to use a singular or plural noun. Students need to see correct usage.

```
3100  IF SA$ = A$(RQ) THEN  PRINT : PRINT  TAB(
      17);"CORRECT":RI = RI + 1:CQ(RQ) = 1: GOTO 3250
3425  REM PRINT SCORE
3450  IF N > 1 THEN  HOME : VTAB 6: PRINT  TAB( 8);"YOU TRIED
      "N" QUESTIONS AND": VTAB 9: PRINT  TAB( 13);"HAVE "RI"
      CORRECT.": GOTO 3460
3455  HOME : VTAB 6: PRINT  TAB( 8);"YOU TRIED "N" QUESTION
      AND": VTAB 9: PRINT  TAB( 13);"HAVE "RI" CORRECT."
```

6. If the student misses a question, not only should the correct answer be given, but the student should be required to enter it. This routine does this as well as centering the correct answer.

```
3205  PRINT : PRINT  TAB( 10);"THE CORRECT ANSWER IS"
3210  LL =  LEN (A$(RQ)): PRINT : PRINT  TAB( ((40 - LL) / 2) +
      1);A$(RQ)
3215  PRINT : PRINT  TAB( 9);"ENTER THE CORRECT ANSWER"
3216  PRINT
3217  PRINT "====> ";: CALL INPT,SA$
3220  IF SA$ <  > A$(RQ) THEN  HTAB 1: CALL  - 998: CALL  -
      868: GOTO 3217
3225  PRINT : PRINT  TAB( 18);"RIGHT"
```

7. The program can also be improved if, when the student completes all the questions, the opportunity to repeat just the questions missed is offered. This can be done with the addition of the following.

```
1900  PRINT N;". ";:X$ = Q$(RQ): GOSUB 9000
2800  PRINT : PRINT "ANSWER =====> ";: CALL INPT,SA$
3100  IF SA$ = A$(RQ) THEN  PRINT : PRINT  TAB(
      17);"CORRECT":RI = RI + 1:CQ(RQ) = 1: GOTO 3250
3280  IF RF THEN  HOME : PRINT : RETURN
4000  REM REPEAT WRONG QUESTIONS
4020  HOME :RI = 0:N = 0
4040  FOR QQ = 1 TO NU
4060  IF CQ(QQ) = 1 THEN 5000
4080  N = N + 1:RQ = QQ:RF = 1: GOSUB 1900
5000  NEXT QQ
5020  GOTO 3450
8989  END
```

A listing of a complete program which incorporates all of the above seven suggestions is at the end of this paper.

MULTIPLE CHOICE QUESTIONS

To use this technique with multiple choice questions, a fixed procedure must be used in writing the question file. The following is an example of such a fixed procedure I have used:

1. Each item allows for the question, five choices, and the correct answer. This is seven parts in all. These will be entered in the program as two-dimensional arrays.

2. Each part must end with a press of the return key and returns must not be used at any other time.

3. Some choices may be left blank, but the blanks must be the choices at the end of the question. A blank choice contains nothing except a carriage return.

4. Answers must be given as A, B, C, D, E.

5. The file must end with the word "END" and a return.

By having your program check each item, it is possible to have questions with two, three, four, or five choices. True and false questions can also be included and all of the routines mentioned for use with single response questions can be used including both random questions and random choices.

THE PROGRAM

This listing uses all of the suggestions in a program for single response questions. It is not in finished professional form, but should work well.

THIS PROGRAM IS NOT TO BE CONSIDERED AS PUBLIC DOMAIN. THOSE ATTENDING THIS PRESENTATION ARE GIVEN PERMISSION TO USE THE PROGRAM FOR THEIR PERSONAL USE IF CREDIT IS GIVEN TO THE AUTHOR

```
900  REM **********************
901  REM *                      *
902  REM *    QUESTIONS &       *
903  REM *      ANSWERS         *
904  REM *                      *
905  REM * PROGRAM INCLUDES     *
906  REM * RANDOM QUESTIONS     *
907  REM *   SCORE KEEPER       *
908  REM * QUIT OR CONTINUE     *
909  REM * RIGHT ANS NEEDED     *
910  REM *  INPUT ANYTHING      *
911  REM * WHOLE WORD PRINTR    *
912  REM * REPEAT WRONG QUES    *
915  REM *  BY LYNN OBERLIN     *
917  REM *     (C) 1986         *
918  REM *                      *
920  REM **********************
999 :
1200  TEXT : HOME
1250  GOSUB 10000
1310  DIM Q$(30),A$(30),QU(30),CQ(30)
1320  D$ =  CHR$ (4):
1322  INPUT "FILE NAME =======> ";F$
1325  PRINT D$"VERIFY"F$
1330  PRINT D$"OPEN"F$
1340  PRINT D$"READ"F$
1400  I = I + 1: CALL INPT,Q$(I): IF Q$(I) <  > "END" THEN  CALL
      INPT,A$(I): GOTO 1400
1405  NU = I - 1
1410  PRINT D$"CLOSE"
1450  HOME
1500  FOR N = 1 TO NU
1550  REM SELECT IN RAMDOM ORDER
1600  RQ =  INT ( RND (1) * NU) + 1
1700  IF QU(RQ) = 1 THEN 1600
1800  QU(RQ) = 1
1900  PRINT N;". ";:X$ = Q$(RQ): GOSUB 9000
2800  PRINT : PRINT "ANSWER =====> ";: CALL INPT,SA$
3100  IF SA$ = A$(RQ) THEN  PRINT : PRINT  TAB( 17);"CORRECT":RI =
      RI + 1:CQ(RQ) = 1: GOTO 3250
3200  PRINT : PRINT  TAB( 18);"WRONG"
3205  PRINT : PRINT  TAB( 10);"THE CORRECT ANSWER IS"
```

```
3210  LL =  LEN (A$(RQ)): PRINT : PRINT  TAB( ((40 - LL) / 2) +
      1);A$(RQ)
3215  PRINT : PRINT  TAB( 9);"ENTER THE CORRECT ANSWER"
3216  PRINT
3217  PRINT "====> ";: CALL INPT,SA$
3220  IF SA$ <  > A$(RQ) THEN  HTAB 1: CALL  - 998: CALL  - 868:
      GOTO 3217
3225  PRINT : PRINT  TAB( 18);"RIGHT"
3250  VTAB 24: HTAB 2: PRINT "ANY KEY TO CONTINUE --  'Q' TO QUIT.
      ";
3275  GET K$: IF K$ = "Q" THEN 3450
3280  IF RF THEN  HOME : PRINT : RETURN
3300  HOME : PRINT : NEXT N:N = N - 1
3425  REM PRINT SCORE
3450  IF N > 1 THEN  HOME : VTAB 6: PRINT  TAB( 8);"YOU TRIED "N"
      QUESTIONS AND": VTAB 9: PRINT  TAB( 13);"HAVE "RI"
      CORRECT.": GOTO 3460
3455  HOME : VTAB 6: PRINT  TAB( 8);"YOU TRIED "N" QUESTION AND":
      VTAB 9: PRINT  TAB( 13);"HAVE "RI" CORRECT."
3460  IF N - RI > 1 AND N = NU THEN  VTAB 11: HTAB 4: PRINT "ANY
      KEY TO REPEAT ITEMS MISSED --": PRINT : HTAB 19: PRINT "'Q'
      TO QUIT. ";: GET K$: PRINT K$: IF K$ <  > "Q" THEN 4000
3470  IF N > RI AND N = NU THEN  VTAB 11: HTAB 4: PRINT "ANY KEY
      TO REPEAT ITEM MISSED --": PRINT : HTAB 19: PRINT "'Q' TO
      QUIT. ";: GET K$: PRINT K$: IF K$ <  > "Q" THEN 4000
3500  VTAB 15: HTAB 17: PRINT "GOOD-BYE"
3550  VTAB 18: HTAB 13: PRINT "HAVE A GOOD DAY"
3560  END
4000  REM REPEAT WRONG QUESTIONS
4020  HOME :RI = 0:N = 0
4040  FOR QQ = 1 TO NU
4060  IF CQ(QQ) = 1 THEN 5000
4080  N = N + 1:RQ = QQ:RF = 1: GOSUB 1900
5000  NEXT QQ
5020  GOTO 3450
8989  END
8990  REM WHOLE WORD PRINTER
9000  D =  PEEK (36): IF  LEN (X$) < 40 - D THEN 9050
9010  C = 39 - D
9020  IF  ASC ( MID$ (X$,C,1)) = 32 OR  ASC ( MID$ (X$,C,1)) = 45
      THEN 9040
9030  C = C - 1: GOTO 9020
9040  PRINT  LEFT$ (X$,C):X$ =  RIGHT$ (X$, LEN (X$) - C): GOTO
      9000
9050  PRINT X$: RETURN
9998  REM INPUT ANYTHING
9999  REM CALL INPT,ST$
10000 INPT = 768: FOR J = INPT TO INPT + 27: READ K: POKE J,K:
      NEXT : DATA
      32,190,222,32,227,223,32,108,221,133,133,132,134,32,111,253,3
      2,57,213,200,162,141,32,233,227,76,154,218
10001 RETURN
```